



Listado con Statement #1

Transcripción

[00:00] Hola, ahora que importamos el proyecto en nuestro workspace, vamos a dar vida a la aplicación. Lo primero que vamos a hacer aquí es cargar la tabla con el listado de productos que ahora se encuentra vacía, entonces vamos a ver en el proyecto cómo es que se carga esta tabla.

[00:16] Todo empieza aquí, en la clase ControlDeStockMain, cuando es instanciado el objeto. Acá en el ControlDeStockFrame, nosotros vamos a ver qué pasa adentro de esta lógica del constructor, entonces para entrar al método nosotros hacemos un F3 acá en el teclado, Eclipse nos lleva justamente para la implementación de este método.

[00:37] Acá en este método tenemos algunos otros métodos que llamo una superclase, la clase padre de esta clase, instanciamos algunos objetos de controller, instanciamos el container y tenemos algunos metros de acá de configuración de campos de formulario, configurar la tabla de contenido y configurar acciones del formulario.

[00:55] Bueno, vamos a entrar aquí a configurar la tabla de contenido. Entrando a este método, tenemos alguna lógica más acá arriba y tenemos este método de cargar tabla en el que vamos a entrar. Entramos aquí a cargar tabla y tenemos que tenemos una variable de productos que recibe el resultado del productoController.listar y después tenemos una lógica acá que está comentada con to do que nosotros tenemos que completar después.

[01:24] Pero vamos a enfocarnos acá en este listar, porque me parece que es acá que vamos a tener que trabajar, entonces entrando acá en listar, tenemos acá un método con el comentario to do también, entonces es aquí que vamos a estar trabajando. Entonces, aquí en el método listar es en donde vamos a estar haciendo toda la lógica para conectar a la base de datos.

[01:46] Si vamos a la consola de MySQL y hacemos el comando `select * from producto`, vemos que tenemos dos productos registrados ahí en nuestra tabla y estos son los productos que queremos mostrar en nuestra aplicación. Bueno, volviendo acá al Eclipse. ¿Se acuerdan que vimos que para acceder a una base de datos todo empieza con la apertura de una conexión?

[02:09] Bueno, entonces vamos a utilizar lo que aprendimos en la clase anterior para poder hacer la conexión con la base de datos. Acá borramos este to do y lo que tenemos que hacer es declarar un `Connection` del paquete `Java.sql`, `Connection con = DriverManager.getConnection`, con los tres parámetros que son la URL de conexión, el usuario y la contraseña.

[02:39] Bueno, esos valores vimos en la clase anterior, pero vamos a recordar acá también. La conexión empieza con `"jdbc:mysql://localhost/control_de_stock?"` Ponemos interrogación acá y vamos a agregar los parámetros de `TimeZone` acá a la conexión de la base de datos, que va a ser `useTimeZone=true&serverTimeZone=UTC`", listo.

[03:22] Ahora vamos para el usuario que es "root" y la contraseña que es "root1234". Ahí está y siempre que aprendimos, siempre que abrimos una conexión, un recurso, nosotros tenemos que cerrarla después de utilizarla. Entonces ya, para no olvidar, ponemos aquí `connection.close`, bueno, pero esto ya sabemos que funciona. La conexión abre y se cierra.

[03:48] Pero ahora nosotros queremos realizar la operación de `select` en la base de datos, para listar los productos y así poblar la tabla de la aplicación. Este

comando select y los otros que existen en SQL son considerados como statements acá en Java.

[04:04] Entonces para crear un statement nosotros debemos utilizar esta conexión que nosotros recién creamos, y nosotros vamos a ejecutar un comando llamado `createStatement`, que nos devuelve un objeto del mismo tipo, del tipo `statement` de acá también. O sea, tenemos un `Statement statement` = el resultado que la conexión hace con la creación de un `statement`.

[04:37] Con este objeto, nosotros vamos a ejecutar nuestra query, la query del select, y para eso nosotros vamos a llamar un otro método del `statement` que va a hacer el `statement` de acá `.execute`. Y el `execute` tiene un parámetro del tipo `string`, que es en donde vamos a estar agregando la query del SQL en el parámetro.

[04:58] Y para esta query, ¿qué queremos hacer? Nosotros queremos todos los datos de producto, pero para no poner acá `select *` vamos a hacer lo siguiente, vamos a dejarla un poco más de explícita, vamos a decir `"SELECT ID, NOMBRE, DESCRIPCION, CANTIDAD FROM PRODUCTO"`. Este es nuestra query, ahí está.

[05:24] Parecido con lo que hicimos en la consola de MySQL, pero ahora estamos haciendo dentro de Java, pero es la misma cosa prácticamente. Ahora nosotros vamos a tomar el resultado de esta ejecución y vamos a agregar una variable que creo que va a ser un listado. Vamos a dejar que Eclipse nos diga qué va a hacer.

[05:42] Voy a decir acá `assign statement to a new local variable`. Y nos dice acá: `"boolean"`. ¿Pero cómo un boolean? Debería ser un listado, no sé, un listado de `string`, por ejemplo. ¿Pero cómo que nos devuelve un boolean? Qué raro eso. ¿Por qué estamos recibiendo este tipo de resultados si esperamos un listado? Bueno, ¿acá qué pasa?

[06:05] Nosotros cuando trabajamos con base de datos, utilizamos muchos tipos de comandos SQL para buscar y manipular informaciones en la base. Los

comandos que más conocemos son el select, que estamos haciendo acá ahora, el insert, el update y el delete. Esos son los más conocidos, los más utilizados.

[06:24] ¿Pero y por qué el execute nos devuelve un boolean acá? ¿Qué sentido tiene? Bueno, este comando devuelve un boolean para indicar que el resultado de este statement de acá, el statement que nosotros creamos y que ejecutamos, este statement nos indica si el resultado es un listado o no. O sea, si el resultado es un listado como acá en el select nos devuelve un true, pero si nosotros estamos ejecutando un insert, un update o un delete, el resultado no va a ser un listado.

[06:57] Entonces nos devuelve el boolean con el valor false. Para comprobar eso vamos a hacer lo siguiente. Vamos a tomar esta variable result acá vamos a hacer un `system.out.println` para imprimir en la consola sus resultados. Así vamos a ver si lo que estoy diciendo tiene sentido. Bueno, vamos a guardar todo acá y vamos a ejecutar esta aplicación.

[07:20] Pero antes de ejecutarla, acá Eclipse se está quejando desde el inicio, y nos dice que tenemos una SQL exception para manejar. ¿Qué vamos a hacer acá? Vamos a hacerle un `throws` primero, acá un `throws` y vamos a pasar la responsabilidad de manejar esta excepción lanzada en el carga tabla que vamos a agregar eso adentro de un bloque de try catch.

[07:46] Y este try catch acá, bueno, el teclado no me está ayudando mucho hoy. Ahora sí, try catch `SQLException`, y nosotros vamos a dar encapsularla adentro de una `RuntimeException`.

[08:05] Entonces vamos a hacer un throw de una new `RuntimeException`, que es una excepción no chequeada, entonces nosotros no tenemos que declarar que el método cargar tabla hace un `throws` de `RuntimeException` para que nosotros podamos con esto encapsulado nos muestra la consola y nosotros no tenemos que estar mandando el `throws` hasta la última parte que hace la llamada.

[08:34] Ahí está, ahora sí, vamos a ejecutar la aplicación y vamos a ver qué dice en la consola. Ahí se levantó la aplicación y tenemos el resultado en la consola con el valor true. O sea, tenemos un listado como resultado, ahora tenemos que descubrir cómo vamos a tomarlo como resultado de hecho.